



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
[www.uspto.gov](http://www.uspto.gov)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/607,397	06/30/2000	Damon Barry	13768.132	9886
47973 7590 08/29/2008 WORKMAN NYDEGGER/MICROSOFT 1000 EAGLE GATE TOWER 60 EAST SOUTH TEMPLE SALT LAKE CITY, UT 84111				
			EXAMINER KISS, ERIC B	
			ART UNIT 2192	PAPER NUMBER
			MAIL DATE 08/29/2008	DELIVERY MODE PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

---

BEFORE THE BOARD OF PATENT APPEALS  
AND INTERFERENCES

---

*Ex parte* DAMON BARRY, GREG VEITH, BRENT JENSEN, and  
FRANK TRUONG

Appeal 2008-0837  
Application 09/607,397<sup>1</sup>  
Technology Center 2100

---

Decided: August 29, 2008

---

Before JOSEPH L. DIXON, JEAN R. HOMERE, and  
ST. JOHN COURTENAY III, *Administrative Patent Judges*.

HOMERE, *Administrative Patent Judge*.

DECISION ON APPEAL

I. STATEMENT OF CASE

Appellants appeal under 35 U.S.C. § 134 from the Examiner's final rejection of claims 1 through 4, 7, 10 through 17, and 20 through 28.

---

<sup>1</sup> Filed on June 30, 2000. The real party in interest is Microsoft Corp.

Claims 5, 6, 8, 9, 18, and 19 have been canceled. We have jurisdiction under 35 U.S.C. § 6(b). We affirm.

Appellants invented a system for selecting and organizing individual test cases into a hierarchy of tests to be run in a software package in order to identify erroneous logic therein. (Spec. 2, 4.) As depicted in Figure 3, the test cases are organized in a three-tiered hierarchy. The first tier includes a test module (76). The second tier includes a plurality of test suites (78) clustered as a module. The third tier in turn includes a plurality of individual test cases (82) clustered in each test suite (78). (Id. 4, 15.) Upon receiving from a user a set of instructions specifying a particular program module to be executed, the system selects a corresponding test hierarchy to test the software package. (Id.)

Independent claim 1 further illustrates the invention. It reads as follows:

1..A computer system for selecting and organizing individual test cases for use in testing a computer program to ensure that the program processes as intended, the system comprising:

one or more program modules storing one or more available test cases, each comprising a set of instructions for testing a feature of the computer program through a language and format independent interface;

a harness client comprising a set of instructions that (i) receives user input specifying one or more filenames corresponding to the one or more program modules, (ii) executes a connector to scan for and discover the one or more available test cases that are stored in the one or more program modules and to organize the one or more available test cases into a test case

hierarchy in which one or more test cases comprise a test suite, and in which one or more test suites comprise a test module, and (iii) receives user input indicating which of the one or more available test cases in the test case hierarchy are selected to be executed on the computer program;

a harness comprising a set of instructions that (i) receives the test case hierarchy, (ii) traverses the test case hierarchy, and (iii) executes each of the one or more available test cases that is selected to be executed on the computer program using the corresponding language and format independent interface of the selected test case to ensure that the computer program processes as intended;

a connector comprising a set of instructions that (i) scans for the one or more available test cases stored in the one or more program modules, (ii) organizes the one or more available test cases into the test case hierarchy by extracting the one or more available test cases from the one or more program modules, and (iii) selectively integrates an interface between the test case hierarchy and the harness regardless of the language or format in which the one or more available test cases were written; and

a processor for executing each selected test case, the harness, the harness client, and the connector.

The Examiner relies on the following prior art:

Hartmann	US 6,505,342 B1	Jun. 7, 2003
		(filed May 31, 2000)

Test Environment Toolkit, *TETware User Guide, Revision 1.2*, (TET\_UG), The Open Group, TET3-UG-1.2 (1998).

Test Environment Toolkit, *Release Notes for TETware Release 3.3*, (TET\_RN), The Open Group, TET3-RN-3.3 (1998).

Test Environment Toolkit, *TETware Programmers Guide, Revision 1.2*, (TET\_PG), The Open Group, TET3-PG-1.2 (1998).

The Examiner rejects the claims on appeal as follows:

- A. Claims 1, 2, 4, 7, 10 through 17, and 20 through 28 stand rejected under 35 U.S.C. § 102(b) as being anticipated by TETware Release 3.3 (TET), including TETware User Guide, Revision 1.2 (TET\_UG) and TETware Programmer's Guide, Revision 1.2 (TET\_PG).
- B. Claim 3 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over the combination of TETware and Hartmann

#### FINDINGS OF FACT

The following findings of fact (FF) are supported by a preponderance of the evidence.

##### *TETware*

1. TETware discloses a test suite structure, which is the largest grouping of test cases to be processed by the TETware controller. The test case, in turn, is the smallest test program unit that can be built or cleaned by the test case controller (tcc). Each test case includes a plurality of invocable elements, which are the smallest program units that can be executed by the tcc. (TET\_UG, p.3, section 2.2.)
2. As depicted in Figure 17, TETware discloses a multi-tiered directory hierarchy including a top tier (\$TET\_ROOT). Further, the directory hierarchy includes in its second tier below the \$TET\_ROOT

directory a test suite root directory (C-API) containing a plurality of test suites (ts). Additionally, the hierarchy directory includes in its third tier a plurality of test cases (chmod, fileno, stat, uname), each having a plurality of invocable elements. (TET\_PG, pp. 148-149.) (See also TET\_PG, p. 3, section 2.3.) (See also TET\_PG, p. 147, section 14.2 defining a test suite residing in subdirectories of tet-root.)

3. TETware discloses that, by default, a test case controller (tcc) looks for test cases below test suite root directory. Alternatively, the tcc looks for the test cases below a specified directory. (TET\_UG, p. 24, section 5.2.6.)

4. TETware discloses that before a user invokes the tcc, the user must ensure that the value of TET\_ROOT environment variable points to the tet root directory on the local system. (TET\_UG, p. 39, section 6.2.2.)

5. TETware also discloses that when the tcc is invoked, it processes each test case within a specified scenario. The tcc processes test cases in the operational mode currently in force. In other words, if the tcc is invoked in the build mode, it builds each test case in the scenario. Similarly, if it is invoked in the build and execute mode, it first builds the test case scenario before it executes it. (TET\_UG, p. 39, section 6.2.3.1, TET\_PG, p. 3, section 2.2.)

6. TETware additionally discloses a scenario file containing scenarios for a test suite. Each scenario file includes a name, which is specified by means of the tcc command line option thereby designating

which test case within a suite is to be executed. (TET\_UG, pp. 26-33, sections 5.3.2-5.3.3.5, TET\_PG, pp. 27-30, sections 4.1-4.2.5.1.)

## PRINCIPLES OF LAW

### ANTICIPATION

In rejecting claims under 35 U.S.C. § 102, “[a] single prior art reference that discloses, either expressly or inherently, each limitation of a claim invalidates that claim by anticipation.” *Perricone v. Medicis Pharmaceutical Corp.*, 432 F.3d 1368, 1375 (Fed. Cir. 2005), citing *Minn. Mining & Mfg. Co. v. Johnson & Johnson Orthopaedics, Inc.*, 976 F.2d 1559, 1565 (Fed. Cir. 1992). “Anticipation of a patent claim requires a finding that the claim at issue ‘reads on’ a prior art reference.” *Atlas Powder Co. v. IRECO, Inc.*, 190 F.3d 1342, 1346 (Fed Cir. 1999) (“In other words, if granting patent protection on the disputed claim would allow the patentee to exclude the public from practicing the prior art, then that claim is anticipated, regardless of whether it also covers subject matter not in the prior art.”) (internal citations omitted).

### OBVIOUSNESS

Appellants have the burden on appeal to the Board to demonstrate error in the Examiner’s position. *See In re Kahn*, 441 F.3d 977, 985-86 (Fed. Cir. 2006) (“On appeal to the Board, an applicant can overcome a rejection [under § 103] by showing insufficient evidence of *prima facie*

obviousness or by rebutting the *prima facie* case with evidence of secondary indicia of nonobviousness.”) (quoting *In re Rouffet*, 149 F.3d 1350, 1355 (Fed. Cir. 1998)).

Section 103 forbids issuance of a patent when “the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.”

*KSR Int'l Co. v. Teleflex Inc.*, 127 S. Ct. 1727, 1734 (2007).

The question of obviousness is resolved on the basis of underlying factual determinations including (1) the scope and content of the prior art, (2) any differences between the claimed subject matter and the prior art, (3) the level of skill in the art, and (4) wherein evidence, so-called secondary considerations. *Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966). *See also KSR*, 127 S. Ct. at 1734 (“While the sequence of these questions might be reordered in any particular case, the [*Graham*] factors continue to define the inquiry that controls.”)

“The combination of familiar elements according to known methods is likely to be obvious when it does no more than yield predictable results.” *Leapfrog Enter., Inc. v. Fisher-Price, Inc.*, 485 F.3d 1157, 1161 (Fed. Cir. 2007) (quoting *KSR Int'l v. Teleflex, Inc.*, 127 S. Ct. 1727, 1739-40 (2007)). “One of the ways in which a patent's subject matter can be proved obvious is by noting that there existed at the time of invention a known problem for



which there was an obvious solution encompassed by the patent's claims.”  
*KSR*, 127 S. Ct. at 1742.

The reasoning given as support for the conclusion of obviousness can be based on interrelated teachings of multiple patents, the effects of demands known to the design community or present in the marketplace, and the background knowledge possessed by a person having ordinary skill in the art. *KSR*, 127 S. Ct. at 1740-41. *See also Dystar Textilfarben GmbH & Co. Deutschland KG v. C.H. Patrick Co.*, 464 F.3d 1356, 1368 (Fed. Cir. 2007).

#### ANALYSIS

Independent claim 1 recites in relevant part the following limitations:

... a harness client comprising a set of instructions that (i) receives user input specifying one or more filenames corresponding to the *one or more* program modules, (ii) executes a connector to scan for and discover the *one or more* available test cases that are stored in the *one or more* program modules and to organize the *one or more* available test cases into a test case hierarchy in which *one or more* test cases comprise a test suite, and in which *one or more* test suites comprises a test module . . . (Emphasis added.)  
(Claims Appendix.)

Appellants argue that TETware does not teach these limitations. (App. Br. 9-13, Reply Br. 2-6.) Particularly, Appellants argue that TETware teaches a controller for executing specified test cases within a pre-existing test case hierarchy. Appellants argue, however, that TETware does not teach a controller that scans a program module to discover available tests

therein, and to organize them in a test case hierarchy comprising of a test module having one or more test suites, each which in turn, comprises one or more individual test cases. (*Id.*)

In response, the Examiner avers that TETware's disclosure of the test case hierarchy including a root node, and a plurality of test case suites containing individual test cases teaches the claimed limitations. (Ans. 10-13.)

Therefore, the pivotal issue before us is whether TETware's disclosure teaches a controller that scans a program module to discover an available test case stored therein, and to subsequently organize the test case in hierarchy comprising of a test module having a test suite, which in turn, includes an individual test case, as recited in independent claim 1. We answer this inquiry in the affirmative.

We begin by considering the scope and meaning of the afore-cited limitations, which must be given their broadest reasonable interpretation consistent with Appellant's disclosure, as explained in *In re Morris*, 127 F.3d 1048, 1054 (Fed. Cir. 1997):

[T]he PTO applies to the verbiage of the proposed claims the broadest reasonable meaning of the words in their ordinary usage as they would be understood by one of ordinary skill in the art, taking into account whatever enlightenment by way of definitions or otherwise that may be afforded by the written description contained in the applicant's specification.

*Id.* at 1054. See also *In re Zletz*, 893 F.2d 319, 321 (Fed. Cir. 1989) (stating that "claims must be interpreted as broadly as their terms reasonably allow."

Appellants' Specification states the following:

In the disclosure and in the claims a “test suite” refers to a set of *one or more* test cases and a “test module” refers to a set of *one or more* test suites. [Emphasis added.]  
(Spec. 8, ll. 16-17.)

Our reviewing court further states, “the ‘ordinary meaning’ of a claim term is its meaning to the ordinary artisan after reading the entire patent.”  
*Phillips v. AWH Corp.*, 415 F.3d 1303, 1321 (Fed. Cir. 2005.)

Upon reviewing Appellants' Specification, we find that a “test module” includes at least one “test suite,” which in turn includes at least one test case. We also note that independent claim 1 recites this same alternative language. Therefore, we find that the claimed limitations in question broadly but reasonably *read on* scanning a program module to discover a single test case, and to organize the test case into a hierarchy having a single test module, including a single test suite, which in turn includes a single test case. *See Atlas Powder* 190 F.3d at 1346. In a nutshell, we construe the cited claimed limitations to broadly but reasonably read on scanning the program module for a single test case arranged as a hierarchy of one. Our broad but reasonable construction is fully consistent with Appellants' Specification, as discussed *supra*.

Claims 1, 2, 4, 7, 10-17, and 20-28

As set forth in the Findings of Fact section, TETware discloses a tcc that builds a designated test case when it operates in the build mode. (FF.

5.) Further, TETware discloses a test case hierarchy having a root node directory, which includes a plurality of test case suites, each of which including in turn a plurality of test cases. (FF. 2.) Additionally, TETware discloses that in the execute mode, the tcc executes designated test cases in the test case hierarchy. (FF. 5-6.) In the above limited situation, we find that when the tcc is operating in the build mode, it creates at least one designated test case. Therefore, tcc creates a hierarchy of one test case in the build mode. Further, in the above limited situation, in the build mode, tcc generates the hierarchy of a plurality of test cases that may be clustered as test cases suites, which in turn may be clustered as a test case module. Therefore, TETware's disclosure teaches scanning a program module to create a hierarchy of a single test. In other words, TETware teaches scanning a program module for a single test arranged as a hierarchy of one. Therefore, we agree with the Examiner's initial showing in the limited situation, and find that the Examiner has set forth a sufficient initial showing of anticipation of independent claim 1. It follows that Appellants have not shown that the Examiner erred in finding that TETware anticipates independent claim 1.

Appellants do not provide separate arguments with respect to the rejection of claims 2, 4, 7, 10 through 17, and 20 through 28. Consequently, these claims fall together with independent claim 1. 37 C.F.R. § 41.37(c)(1)(vii).

### Claim 3

Appellants argue that Hartmann does not cure the deficiencies of the TETware reference. (App. Br. 14.) As discussed above, we find so such deficiencies in TETware for Hartmann to remedy. It follows that Appellants have not shown that the Examiner erred in concluding that the combination of TETware renders claim 3 unpatentable.

### CONCLUSIONS OF LAW

Appellants have not shown that the Examiner erred in finding that claims 1, 2, 4, 7, 10 through 17, and 20 through 28 are anticipated under 35 U.S.C. § 102. Further, Appellants have not shown that the Examiner erred in concluding that claim 3 is unpatentable under 35 U.S.C. § 103.

### DECISION

We affirm the Examiner's decision rejecting claims 1 through 4, 7, 10 through 17, and 20 through 28.

Appeal 2008-0837  
Application 09/607,397

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED

rwk

WORKMAN NYDEGGER/MICROSOFT  
1000 EAGLE GATE TOWER  
60 EAST SOUTH TEMPLE  
SALT LAKE CITY UT 84111